```
//[城市正視圖/Urban Elevations]
#define IN "P0514IN.txt"
#define OUT "P0514OUT.txt"
//****************************************
#include <iostream>
#include <time.h>
using namespace std;
void redir(void);
//****************************************
/* Work Space*/
#include <algorithm>
struct Building{
    int id;
    double x, y, w, d, h;//(x, y), width, depth, height

    bool operator < (Building &rhs) { // *this < rhs
        return x < rhs.x || (x == rhs.x && y < rhs.y);
    }
}b[100]; //根據題意
double x[100*2];
int n;

bool visible(int i, double mx);
bool cover(int i, double mx);
//****************************************
int main(void)
{
    redir(); //redirection
//****************************************
/* Work Space*/
    int i, j;
    int m;
    int kase = 0;
    bool vis;

    while(scanf("%d", &n) == 1 && n){
        for(i=0; i<n; i++){
            b[i].id = i+1;
            scanf("%lf%lf%lf%lf%lf", &b[i].x, &b[i].y, &b[i].w, &b[i].d,  &b[i].h);
            x[i*2] = b[i].x;
            x[i*2+1] = b[i].x + b[i].w;
        }
        sort(b, b+n); //將建物排序

        sort(x, x+n*2); //劃分出區間
        m = unique(x, x+n*2) - x; //x座標排序後去除重複，得到m個座標

        if(kase++){
            printf("\n");
        }
        printf("For map #%d, the visible buildings are numbered as follows:\n%d", kase, b[0].id);
        for(i=1; i<n; i++){
            vis = false;
            for(j=0; j<m-1; j++){
                if(visible(i, (x[j]+x[j+1])/2)){
                    vis = true;
                    break;
                }
            }
            if(vis){
                printf(" %d", b[i].id);
            }
        }
        printf("\n");
    }
//****************************************
    freopen("CON", "r", stdin); //取消重新導向
    freopen("CON", "w", stdout);

    printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間

    system("pause");
    return 0; //the end...
}

void redir(void)
{
    freopen(IN, "r", stdin);
    freopen(OUT, "w", stdout);
}
//****************************************
/* Work Space*/
//判斷建築物i在x=mx處是否可見
bool visible(int i, double mx)
{
    int k;

    if(!cover(i, mx)){
        return false;
    }

    for(k=0; k<n; k++){
        if(b[k].y < b[i].y && b[k].h >= b[i].h && cover(k, mx)){
            return false;
        }
    }

    return true;
}

//判斷建築物i是否包含這個mx區間
bool cover(int i, double mx)
{
    return  b[i].x <= mx && mx <= (b[i].x+b[i].w);
}

//Input(IN) Sample
/*
14
160 0 30 60 30
125 0 32 28 60
95 0 27 28 40
70 35 19 55 90
0 0 60 35 80
0 40 29 20 60
35 40 25 45 80
0 67 25 20 50
0 92 90 20 80
95 38 55 12 50
95 60 60 13 30
95 80 45 25 50
165 65 15 15 25
165 85 10 15 35
0
*/
```